**Software Requirements Specification**

for

trendAssist App

Version 1.0 approved

**Prepared by:**

Andre Manz

Cade Wall

Aaron Turner

Aaron Riggs

Xujia Wu

Mayur Bhakta

Flying Mongeese

February 5, 2018

Table of Contents

Revision History.............................................................................................................................2

1. Introduction..............................................................................................................................2

1.1 Purpose......................................................................................................................2

1.2 Document Conventions..............................................................................................2

1.3 Intended Audience and Reading Suggestions...........................................................2

1.4 Product Scope...........................................................................................................2

1.5 References.................................................................................................................3

2. Overall Description..................................................................................................................3

2.1 Product Perspective...................................................................................................3

2.2 Product Functions......................................................................................................3

2.3 User Classes and Characteristics..............................................................................3

2.4 Operating Environment..............................................................................................4

2.5 Design and Implementation Constraints....................................................................4

2.6 User Documentation..................................................................................................4

2.7 Assumptions and Dependencies...............................................................................4

3. External Interface Requirements.............................................................................................5

3.1 User Interfaces...........................................................................................................5

3.2 Hardware Interfaces...................................................................................................6

3.3 Software Interfaces....................................................................................................6

3.4 Communications Interfaces.......................................................................................7

4. System Features......................................................................................................................7

4.1 System Feature 1.......................................................................................................7

4.2 System Feature 2 (and so on)...................................................................................8

5. Other Nonfunctional Requirements.........................................................................................8

5.1 Performance Requirements.......................................................................................8

5.2 Safety Requirements.................................................................................................8

5.3 Security Requirements...............................................................................................8

5.4 Software Quality Attributes........................................................................................8

5.5 Business Rules..........................................................................................................9

6. Other Requirements................................................................................................................9

Appendix A: Glossary....................................................................................................................9

Appendix B: Analysis Models........................................................................................................9

Appendix C: To Be Determined List...............................................................................................9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **1. Introduction**

## **1.1 Purpose**

trendAssist is a business finance and trend prediction program that allows the user to input data over sales made in a day; this data will then be submitted into a database containing financial sales data for every day of the week. It will also allow the user to input data in years past to increase the accuracy of sales predictions; the program will then use this information to generate a new estimation of revenue for that same day of the week in the future and describe requirements needed to hit specific profit margins. This will enable the user (owner or manager) to determine the optimal staffing needed on any given day to support the expected crowd and maximize profits. The most recent year’s data will be weighted higher than the previous year’s data to maintain accuracy and adjust for recent trends.

## **1.2 Document Conventions**

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## **1.3 Intended Audience and Reading Suggestions**

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

Small business owners who want to be able to forecast gross sales by using inputted data from past years as reference. The program will also be able to help with scheduling an appropriate amount of employees based on foot traffic.

## **1.4 Product Scope**

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

Our project will primarily focus on helping employers schedule an appropriate amount of employees to work on a given night and project estimate revenues for a specific date.

## **1.5 References**

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# **2. Overall Description**

## **2.1 Product Perspective**

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

This program is a new, self-contained product that will be convenient for small business owners to make use of after inputting financial data by predicting potential future customer traffic based on this data on a given day in the future.

## **2.2 Product Functions**

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

The product will allow the user to create forecasts based on gross revenue. They will also be able to select how far out or for what days they would like to forecast for. The product will give an estimate of how many workers need to be scheduled based off of potential income.

## **2.3 User Classes and Characteristics**

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

* Admin Account: authorization for total accessibility to view and modify all sales data
* Employee Account: authorization to input new sales data

## **2.4 Operating Environment**

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

The software will be able to be ran on Windows OS and mac OS.

Java, MySQL, Windows, macOS.

## **2.5 Design and Implementation Constraints**

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## **2.6 User Documentation**

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

Generic readme.txt file that includes basic instructions on how to navigate the program, input new data, view past data, generate new prediction graph, etc.

## **2.7 Assumptions and Dependencies**

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

The software will mainly depend on the user to have internet connection, basic computer skills, and have a connection to the Database where the data is stored. Pre-existing accounts are assumed to be entered into the database.

Internet connection. Basic computer skills. Database connection.

# **3. External Interface Requirements**

## **3.1 User Interfaces**

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

**SCREEN 1: Login Screen**

The opening screen will be comprised of a login screen that will take the user’s Username and Password.

Text Field 1: Username

Pre-existing account username stored in account mySQL database.

Text Field 2: Password

Password associated with pre-existing account.

Button 1: Sign In

Button to verify login credentials

Once verified, the main page will have buttons for the main functions of the program.

**SCREEN 2: Home Page**

Button 1: Generate Sale Prediction

This button will lead to a page that will prompt the user for a specific date. A small description will remind the user to only input a date that does not predate the current date.

Button 2: Modify Sales Data

This button will lead to another screen which will allow the user to select a specific date to modify the sales data. This is the screen that will be used to input new data after each night.

Button 3: Account Settings

This button will lead the user to a screen which will allow them to modify the Username and Password.

Button 4: Create New Account

This button will lead the user to a page where they will be able to enter a new Username and Password. This will create an account where the new user is authorized to fully utilize all of the functionality of the program.

**SCREEN 3: Data Entry Page**

Field 1: Date of Sales

Field 2: Sales Value

Button 1: Submit Data

**SCREEN 4: Account Creation**

Field 1: Username

Field 2: Password

Field 3: Confirm Password

Button 1: Create My Account

## **3.2 Hardware Interfaces**

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

This program will be lightweight enough to store on a flash drive to maintain portability and be cross-platform with Windows and macOS devices alike to be as accessible as possible. It will not require the user to install on their machine but will be ran locally off the flash drive.

## **3.3 Software Interfaces**

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Splitting the project up into 3 sections, frontend, security/encryption, and backend, the database will be a part of the backend that Python will be able to interface with and pass data to the frontend, written in Java to handle the data and number crunching. The database information will be fully secured and encrypted with HTTPS encryption.

Not fully discussed yet.

Databases: Excel, mySQL

Languages: Python, Java

## **3.4 Communications Interfaces**

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# **4. System Features**

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## **4.1 System Feature 1**

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## **4.2 System Feature 2 (and so on)**

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## **5.2 Safety Requirements**

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## **5.3 Security Requirements**

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

All customer data will be encrypted with AES-256 the authorized users will have user accounts that will be related to AES encryption keys to unlock the customers data when requested by an authorized user. The program will feature a login screen so that an authorized user may log in securely, the user account data will be stored with encrypted keys that will check against secure hashes so that even if someone were to get access to the device they would not have access to the customers login.

## **5.4 Software Quality Attributes**

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## **5.5 Business Rules**

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

Now this is non-profit product.

# **6. Other Requirements**

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>